

High-Precision Pointing on Large Wall Displays using Small Handheld Devices

Mathieu Nancel, Olivier Chapuis, Emmanuel Pietriga, Xing-Dong Yang, Irani Pourang, Michel Beaudouin-Lafon

▶ **To cite this version:**

Mathieu Nancel, Olivier Chapuis, Emmanuel Pietriga, Xing-Dong Yang, Irani Pourang, et al.. High-Precision Pointing on Large Wall Displays using Small Handheld Devices. 31th international conference on Human factors in computing systems, Apr 2013, Paris, France. pp.831-840, 10.1145/2470654.2470773 . hal-00786532v2

HAL Id: hal-00786532
https://hal.science/hal-00786532v2

Submitted on 1 May 2013

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

High-Precision Pointing on Large Wall Displays using Small Handheld Devices

Mathieu Nancel[1,2] **Olivier Chapuis**[1,2] **Emmanuel Pietriga**[2,3] **Xing-Dong Yang**[4]
Pourang Irani[5] **Michel Beaudouin-Lafon**[1,2]

[1]Univ Paris-Sud & CNRS (LRI) Orsay, France
[2]INRIA Orsay, France
[3]INRIA CHILE - CIRIC Santiago, Chile
[4]University of Alberta Alberta, Canada
[5]University of Manitoba Manitoba, Canada

ABSTRACT

Rich interaction with high-resolution wall displays is not limited to remotely pointing at targets. Other relevant types of interaction include virtual navigation, text entry, and direct manipulation of control widgets. However, most techniques for remotely acquiring targets with high precision have studied remote pointing in isolation, focusing on pointing efficiency and ignoring the need to support these other types of interaction. We investigate high-precision pointing techniques capable of acquiring targets as small as 4 millimeters on a 5.5 meters wide display while leaving up to 93% of a typical tablet device's screen space available for task-specific widgets. We compare these techniques to state-of-the-art distant pointing techniques and show that two of our techniques, a purely relative one and one that uses head orientation, perform as well or better than the best pointing-only input techniques while using a fraction of the interaction resources.

ACM Classification Keywords

H.5.2 [Information interfaces and presentation]: User Interfaces. - Graphical user interfaces.

Author Keywords

Wall Displays; Handheld Devices; Pointing

INTRODUCTION

Ultra-high-resolution wall-sized displays enable the interactive visualization of massive datasets [5, 27]. Application domains range from scientific visualization to computer-aided design and command & control centers. Usually made of a mosaic of LCD panels, such displays typically feature a resolution of 100 pixels per inch. Beyond their significantly increased display capacity, they afford a more physical form of navigation than lower-resolution large displays [2]: users can get an overview of the data from a distance, the wall fully fitting in their field of view; and they can see more details about a particular region of interest by stepping closer to the wall.

To fully take advantage of the capabilities of this new generation of wall-sized displays, users must be able to move freely in front of them. They must also be able to interact with them from any location, without being tethered by clumsy equipment. This entails two main requirements, as illustrated in Figure 1: (1) users must be able to remotely select graphical objects, which can be very small and difficult to reach; and (2) users must be able to perform other input actions, e.g., adjust parameters associated with the selected object, enter text or perform virtual navigation.

Portable surfaces have been used in prior work as pads for interacting with virtual environments, e.g., to take notes [30], or as devices to facilitate remote pointing [23]. We argue that portable devices can actually accommodate both types of tasks simultaneously, enabling users to perform truly high-precision remote pointing and to manipulate the objects selected thereby. The device can also act as a secondary display. The type of portable device, e.g., a tablet or a smartphone, and the kind of interface controls and interaction techniques that it should provide are largely application-dependent. As a first step, we focus in this paper on the design and evaluation of high-precision remote pointing techniques (requirement #1) that use only a fraction of the surface of a portable device for pointing (requirement #2).

We introduce a design space for techniques that use a handheld device with an area dedicated to pointing that varies in size, and relative pointing acceleration functions specifically tuned to the particular context of ultra-high-resolution wall-sized displays. Several of these techniques can be coupled with information about the user's natural head movements, which we have observed to be a good predictor of the next target location. We systematically evaluate these techniques, showing that purely relative techniques can be surprisingly efficient when tuned carefully, even with small pointing areas. We also show that even smaller pointing areas can be used with similar performance levels by combining head orientation for coarse pointing with touch-based relative pointing. Our techniques generated very positive user feedback, considering the difficulty of the tasks, and fared as well as state-of-the-art remote pointing techniques while enabling additional interactions through the handheld device.



Figure 1. Interacting with a wall display using a tablet that features both an area dedicated to high-precision pointing for the remote selection of graphical objects on the wall, and control widgets to act on those objects.

Nancel M., Chapuis O., Pietriga E., Yang X.-D., Irani P. and Beaudouin-Lafon M. High-Precision Pointing on Large Wall Displays using Small Handheld Devices. In CHI '13: Proceedings of the SIGCHI Conference on Human Factors and Computing Systems, ACM (2013), 831–840.

© ACM, 2012. This is the author's version of the work. It is posted here by permission of ACM for your personal use. Not for redistribution. The definitive version will be published in *CHI 2013, April 27–May 2, 2013, Paris, France.*

LITERATURE REVIEW

Target acquisition on wall-sized displays. Laser pointers or ray-casting [18, 17] are the most common absolute positioning method for pointing from a distance. However, ray-casting does not adapt well to ultra-high-resolution displays as hand tremor and involuntary motion due to fatigue are amplified when the user moves further away from the display [25]. In a controlled study, Kopper *et al.* [22] showed that the speed of a distal pointing task can be described as a function of the angular amplitude of movement and the angular size of the target. Their model presents two fundamental differences between a distal pointing task and the one modeled by Fitts' law. In distal pointing, the size of the target has a much larger effect on movement time than the amplitude. Unlike a Fitts' pointing task, difficulty of distal pointing grows quadratically, rather than linearly. The pointing model based on angular target size supports recent empirical findings showing that ray-casting is fast with large distant objects but error-prone with smaller items [25, 34]. By contrast, techniques that use relative control [34, 4] enable more precise cursor positioning and easier acquisition of small targets.

Interactions using head tracking. Object selection is often preceded by a visual search for the target. Head orientation provides a good approximation of where a user is looking [33, 12, 28], and can be exploited in conjunction with any pointing device used in the environment [19, 28]. Head movements have been shown to support a variety of interaction techniques [19, 24, 20]. Our approach integrates head orientation with cursor selection. It was inspired by Ashdown *et al.* [1], who use head orientation for positioning a cursor on the monitor of interest, thereby reducing mouse displacements in a multi-monitor setup.

Dual-mode target acquisition. Hybrid pointing consists of combining absolute pointing for rapid cursor movements with relative pointing for accuracy. For example, ray-casting, which uses absolute positioning, can be improved by combining it with relative cursor movements [34, 23, 11, 21, 14]. Adaptive Pointing [21] allows switching between two modes based on the speed of the user's hand movements. A study showed that such a hybrid control mechanism reduces error rates by 63% over ray-casting. Mode-switching is implicit, requiring users to adjust their visual search and pointing actions when the switch occurs [14]. By contrast, explicit mode switching lets users control when to switch between relative and absolute positioning [34, 11]. However, the overhead introduced by the mode switch cancels out what is gained by using ray-casting for rapid movements [34]. Alternatives to explicit mode switching include tracking eye gaze or body motion to anticipate or perform part of the targeting action. Zhai *et al.* [36] developed MAGIC pointing, a technique that uses eye gaze to rapidly move the cursor as close as possible to the target on a single desktop monitor. To finish the pointing task, users then move the cursor to the target with a standard trackpad.

To avoid the cost of explicit mode switching, our approach uses head motion to move the cursor as close as possible to the intended target area. Other techniques can then be used to precisely position the cursor on the target.

Pointing via mobile touch-screens. The use of mobile devices to control distant objects has received much attention in recent years. One approach is to control a remote cursor via the mobile device's touch-screen. For example, ARC-Pad enables both absolute and relative pointing on a large display via the touch-screen of a mobile device [23]. Alternatives include using embedded sensors, such as tilt [31] or the phone's camera [8]. With Touch Projector [8], users aim the mobile device's camera at the display and manipulate its content by touching and dragging content on the small screen. The techniques we have designed use some of these approaches for remote control of the cursor.

Pointing Facilitation Techniques. Pointing time can be improved either by reducing distance to target, increasing target width, or a combination of both. Drag-and-pop [3] and the Vacuum [6] are two examples designed specifically for use on large displays. Such techniques generally rely on prior knowledge of potential targets, while our work focuses on generic, target-agnostic pointing techniques. Combining our approach with pointing facilitation techniques is certainly possible since the latter make few assumptions about the physical input device used, but is left for future work.

TECHNIQUES

We consider contexts of use where the primary purpose of the handheld device is to accommodate widgets for the advanced control of objects selected via pointing. The area dedicated to pointing on the device, or *pointing zone*, should therefore not be too large. However, if too small, pointing will likely be inefficient. Our goal is to identify the best trade-off between screen real-estate allocation and good pointing performance.

Relative Pointing – As mentioned earlier, a simple absolute mapping from the handheld device to a ultra-high-resolution wall display does not work, because one pixel on the handheld maps to several dozen pixels on the wall, even if we were to make the pointing zone full-screen. Relative mapping, i.e., using the pointing zone as a touchpad, enables users to point precisely but requires careful tuning of the pointer acceleration function, which can be a complex and tedious process. The acceleration functions implemented in major operating systems have been parameterized for desktop environments [9]. They work for single- and multi-monitor display configurations, but are not adapted to ultra-high-resolution walls, which typically feature a 200-to-400 inches diagonal and a very high resolution, e.g., $20\ 480 \times 6\ 400$ for the 228-inch (5.79 meters) display wall in Figure 1. Previous work [26] showed that basic relative pointing, if tuned properly, could actually be a viable candidate for high-precision pointing on wall displays. One of our contributions is a method to obtain such optimized transfer functions.

Dual-Precision Pointing – Pointer acceleration functions dynamically adjust the control-display (CD) gain based on the dynamics of users' movements. An alternative approach is to let users explicitly switch between different CD gains, typically between a *Coarse* mode that allows fast repositioning of the cursor across large distances, and a slower but more *Precise* mode that users can engage when they want to adjust the cursor position to acquire a very small target [26].

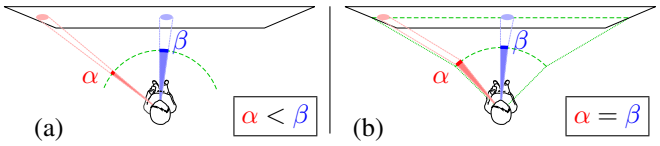


Figure 2. The problem of varying angular motor size with ray-casting (a) can be addressed using an indirect absolute mapping (b).

The main challenge when designing dual-precision techniques is to seamlessly integrate the two modes so as to minimize the cognitive and motor costs of switching. All the techniques presented in this paper assign the *Precise* mode to single-finger drag gestures performed with the dominant hand in the pointing zone. Given our requirements that users should be free to move in front of the display, and since the non-dominant hand will typically hold the device, this leaves two main options for *Coarse* mode control: use the dominant hand in a different configuration, or use a different body part. We experimented with several options to eventually identify two viable candidates through iterative design, prototyping, piloting and tuning: head orientation, and double-finger drag gestures with the dominant hand in the pointing zone. Both approaches are detailed below.

Two-finger pad-based coarse pointing – This technique is inspired by ARC-Pad [23] (Absolute+Relative Cursor Pad), a dual-precision pointing technique that provides users with an absolute and a relative pointing mode. A typical ARC-Pad pointing task is composed of a tap on a touch-sensitive handheld device, followed by a drag gesture. The tap gesture coarsely positions the cursor on the large display according to an absolute mapping of the handheld device's surface to the large display. The following drag gesture is interpreted as relative movements of the cursor to adjust its position. The original technique was designed for, and evaluated on, much lower-resolution large displays than those considered here. It proved very difficult to use with very small targets on ultra-high-resolution wall displays, mainly because the absolute mapping is far too imprecise in that context and often requires several relative-mode drag gestures to adjust the cursor's position, causing much clutching in the second phase. We thus enhanced the original technique and created ARC-Pad^{2F}.

ARC-Pad^{2F} (^{2F} stands for *two fingers*) distinguishes between absolute and relative pointing by the number of fingers involved in performing the pointing gestures rather than the type of gesture performed. A single-finger drag gesture controls the cursor in relative mode (as with the original ARC-Pad); a drag gesture performed with two conjoined fingers is interpreted as absolute positioning of the cursor. In contrast to the original method that used taps for absolute positioning, users can now adjust the cursor position in absolute mode by dragging with two fingers, and then switch to relative mode for more precise, relative adjustments of the cursor position. The switch from absolute to relative mode is triggered whenever at least one of the fingers is lifted from the pad's surface. This means that users can either lift a single finger at the current location and continue dragging, or they can lift both fingers, adjust their hand position relative to the pad, and touch anywhere on the surface with a single finger. This second option can be very useful when pointing at a target near the edges of the display, as it allows users to initiate relative drag gestures from the center of the pointing zone, in any direction.

Head-based coarse pointing – The second viable approach that we identified for *Coarse* control uses the natural movements of the head that occur when remotely pointing at targets on a wall. Confirming prior work [16, 36], we observed that users consistently stabilize their gaze and lock it on the target before acquiring it. While tracking the users' gaze is not a practical option in our context, we found head orientation to be a good coarse indirect indicator of where users are looking[1], and thus a good predictor of the rough location of the next intended target on the wall display.

Our technique controls the cursor's location in *Coarse* mode through a linear interpolation from a comfortable angular operating range to the dimensions of the display. We favored an indirect mapping over direct ray-casting (or laser pointing) for two reasons. First, being perspective-based, ray-casting would have caused targets of the same absolute size to have noticeably different *motor* sizes depending on their location with respect to the user's physical position, as illustrated in Figure 2-a (this effect gets amplified as users get closer to the display). Our mapping allows any area of the display to be reached with equal motor precision. We also wanted to optimize the input operating range, within the limits of comfortable neck positions. As illustrated in Figure 2-b, we addressed these issues by mapping a location-independent, fixed-size angular operating range centered on the orthogonal projection of the user's location on the flat display surface. This ensures that when users move in front of the display, looking straight ahead always sets the cursor exactly in front of them. As they look further away, the cursor is progressively offset from the head's direction (up to 9°), accounting for extra eye rotation. This offset makes it possible to point at targets on the sides of the wall comfortably while maximizing accuracy in the central area.

Cost of Switching – We observed during the early phases of development and pilot testing that users would sometimes hesitate to switch to *Precise* mode. This could happen because they thought that they could complete the task in *Coarse* mode, eventually failing or succeeding but at the cost of longer task completion times. This could also happen because they wanted to get as close as possible to the target before switching to *Precise* mode, sometimes loosing a significant amount of time in doing so for no actual benefit since the time lost in *Coarse* mode was not compensated by larger time savings in *Precise* mode.

This raised a question that we had not anticipated: beyond minimizing the motor cost of switching between modes, is it possible to minimize the *cognitive* cost associated with making the decision to switch between the two modes?

We explored an approach where the cognitive load is transferred to the perceptual system, hypothesizing that this would significantly reduce the mode switching cost. We designed variations of the above techniques that artificially limit the precision of the *Coarse* pointing mode: users approach the

[1] Users rotate their head to look at off-centered targets so as to minimize the effort put on ocular muscles [12].

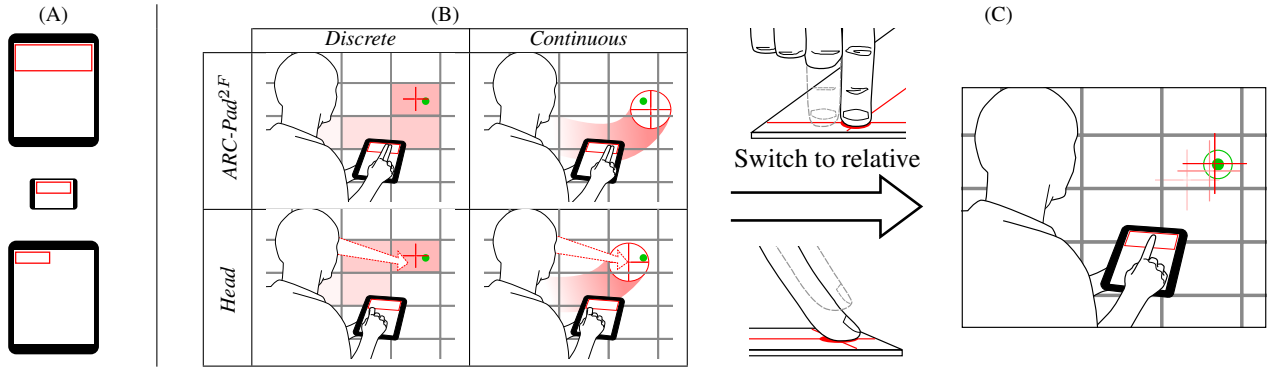


Figure 3. The four dual-precision techniques. (A) The three device / pointing zone pairs used in our experiments: tablet with large zone (Experiments 1 and 3), smartphone with small zone (Experiment 2) and tablet with small zone (Experiment 3). (B) The four coarse (absolute) modes, combinations of *Head* vs. *ARC-Pad*^{2F} and *Discrete* vs. *Continuous*. (C) In all cases, using a single finger in the pointing area switches to precise (relative) mode.

target fast and know when to switch to precise mode simply because there is no other option.

We *discretized* the wall display into cells according to its 8×4 matrix of 30" LCD panels. Using portions of the LCD panels, e.g. halves or quarters, would have made the cells too small for precise acquisition. We also wanted to avoid having cells inconsistently crossing screen bezels. In *Coarse* mode, users can only jump from screen to screen. Jumping to a screen puts the cursor at the center of that screen, and users have to switch to *Precise* mode to reposition the cursor within it. We call this process *discretization* to emphasize that the pointing resolution in *Coarse* mode is artificially degraded while keeping the same physical input resolution.

The resulting technique, when coupled with *Head* control, is somewhat reminiscent of the Rake cursor [7] and similar multi-cursor desktop pointing techniques [32] that use eye gaze to select the active cursor in a matrix of cursors all controlled with the same input device, such as a mouse.

Four dual-precision pointing techniques – We eventually narrowed our design space down to four pointing techniques by combining the two coarse input control techniques (pad-based vs. head-based) with the two coarse mode input precisions described above (*continuous* vs. *discretized*) – see Figure 3. As mentioned earlier, all four techniques, summarized below, use the handheld device's pointing zone to control the cursor in *Precise* mode, with optimized CD gain transfer functions.

- ARC-Pad^{2F} + *Continuous* = *ContPad*: two conjoined fingers in the handheld's pointing zone activate *Coarse* mode (absolute mapping of the zone to the wall); a single finger activates *Precise* mode (relative cursor displacements).
- ARC-Pad^{2F} + *Discrete* = *DiscPad*: same as above but pointing precision in *Coarse* mode is artificially restricted: the cursor can only get positioned at the center of any given LCD panel, requiring a switch to *Precise* mode for any further cursor position adjustments.
- *Head* + *Continuous* = *ContHead*: head orientation provides *Coarse* control of the cursor, without any artificial restriction on pointing precision. Touching the pointing zone on the handheld automatically switches to *Precise* mode (relative cursor displacements).
- *Head* + *Discrete* = *DiscHead*: same as above, but pointing precision in *Coarse* mode is artificially restricted. Head orientation can only make the cursor jump to the center of any given LCD panel, as with *DiscPad*.

Head-based techniques feature a $500ms$ delay after finger release from the handheld's surface before switching back to coarse mode, giving users enough time to clutch or tap.

CONTROL-DISPLAY TRANSFER FUNCTIONS

As mentioned earlier, optimizing the transfer functions that control pointer acceleration in relative mode for ultra-high-resolution wall displays is challenging. In this section, we explain our method for calibrating those functions.

Pointer acceleration consists of applying a transfer function to the Control-Display (CD) gain based on the pointer's dynamics. The literature on CD gain transfer functions for such high-resolution and large display surfaces is sparse. Apart from recent work by Casiez *et al.* [9, 10] that provides a general framework but does not directly address our configuration, the only documented calibration methods are those related to PRISM [13] and its refinements [21, 15]. But these calibration methods were designed to support absolute-to-relative transfer functions for pointing techniques that feature an implicit switch between absolute and relative modes. While we later compare our techniques with the latest developments in this area, those calibration methods are of little use in our context.

All of our techniques use pointer acceleration in precise mode, and some of them in coarse mode as well. The CD gain must be large at high input speeds so the user can traverse the entire screen (15 000 to 30 000 pixels) quickly with no or little clutching. Conversely, the CD gain must be small at low input speeds for high-precision cursor control.

Some operating systems use sigmoid transfer functions [9, 10] that are characterized by a slope that smoothly gets steeper before decreasing again, as illustrated in Figure 4. The lower slopes at each end of the curve enable higher precision at low input velocities and bound cursor speed.

To model such curves, we use a generalized logistic function:

$$CD(x) = \frac{CD_{max} - CD_{min}}{1 + e^{-\lambda(x - V_{inf})}} + CD_{min}$$

where: $V_{inf} = ratio_{inf} \cdot (V_{max} - V_{min}) + V_{min}$. This function can be tuned with six parameters: V_{min} and V_{max} bound the input

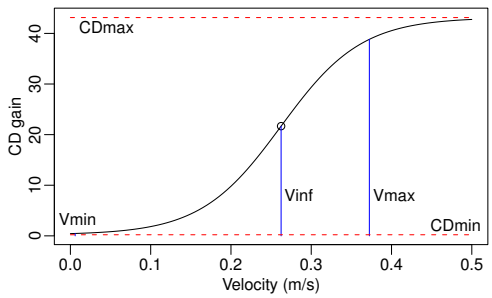


Figure 4. An example of logistic sigmoid curve (large zone *RelaLarge*)

range of the function by specifying lower and upper velocities beyond which accurate control becomes difficult; $ratio_{inf}$ sets the function's inflection point within this range; CD_{min} and CD_{max} are asymptotic values that define the function's output range. Finally, $\lambda \in [0; \infty]$ defines the curve's slope at its inflection point ($\lambda = 0$ yields a constant function, $\lambda = \infty$ a step function). Adjusting λ smoothes the curve.

We tune these parameter values as follows:

1. V_{min} and V_{max} are respectively the 90^{th} quantile and the median of two corpora of velocities corresponding to voluntarily slow and fast finger movements on a tablet;
2. $CD_{min} = \frac{W_{min} \text{or } L}{100 \times \text{res}_{\text{device}}}$ maps the lower bound for the amplitude of input movements that are considered usable for selecting a target (the equivalent of 100 *input* pixels at device resolution $\text{res}_{\text{device}}$) to the minimum target size W_{min} or to the cursor width L [26] for relative *Coarse* modes;
3. The initial value for CD_{max} is $\frac{L}{\min(\text{width}_{input}, \text{height}_{input})}$ or $\frac{A_{max}}{\text{width}_{input}}$, mapping either the smallest dimension of the input zone to the expected *Precise* mode amplitude L [26], or the input zone width to the maximum pointing amplitude A_{max} for mode-less techniques and *Coarse* modes (A_{max} in our experiment corresponds to mostly horizontal movements).
4. $ratio_{inf}$, λ and CD_{max} are adjusted so the transition between the highest and lowest CD gains is as smooth as possible.

The main criterion for deciding that a given function is properly tuned is that users can perform the most difficult task (largest amplitude, smallest target width) and no further parameter adjustment improves pointing performance or the users' subjective perception of smoothness and precision.

Three volunteers were asked to test the techniques in an informal iterative design process of 800 pointing trials per user on average, before reaching a consensus. Table 1 gives the parameter values that we obtained for the two pointing zones that served in the experiments reported below: a large zone that fits within a tablet device in portrait mode, and a small zone that fits within a smartphone in landscape mode.

Generally-speaking, V_{min} and V_{max} have lower values for the small zone because the physical (motor) space dedicated to pointing is smaller. Similarly, the small zone features a higher CD_{max}, allowing for faster cursor movements so as to compensate for the smaller operating range. *Discrete* techniques feature a higher CD_{max} than *Continuous* ones to compensate for the larger distance between the target and the point where the mode switch has to be performed.

Size	Technique	CD_{max}	λ	$ratio_{inf}$
Large zone 148 × 49 (mm)	*ContHead*$_L$	9.9	19.9	0.4
	ContPad$_L$	6.6	17.5	0.4
	DiscHead$_L$	12.9	17.5	0.4
	DiscPad$_L$			
	RelaLarge	43.1	20.0	0.5
Small zone 75 × 26 (mm)	*ContHead*$_S$	6.3	26.5	0.4
	ContPad$_S$			
	DiscHead$_S$	16.4	26.5	0.7
	DiscPad$_S$			
	RelaSmall	83.0	22.5	1.0

Large zone: $V_{min} = 0.006m/s$, $V_{max} = 0.37m/s$, $CD_{min} = 0.22$
Small zone: $V_{min} = 0.003m/s$, $V_{max} = 0.19m/s$, $CD_{min} = 0.27$

Table 1. Transfer function parameter values.

COMPARING THE DUAL-PRECISION TECHNIQUES

We conducted two experiments to evaluate the performance of the four dual-precision pointing techniques introduced above, *ContHead*, *ContPad*, *DiscHead* and *DiscPad*, and to assess the cost of mode switching. The two experiments followed the same design but used different devices: a tablet with a large pointing zone in the first experiment, and a smartphone with a small pointing zone in the second one. For the sake of clarity, we use subscripts ($_L$ for *Large* and $_S$ for *Small*) when referring specifically to either the tablet or smartphone implementation, e.g., *ContHead*$_L$ and *ContHead*$_S$.

Our hypotheses are as follows:

(H1) *Discrete* techniques lower the cognitive (and thus overall) cost of mode switching by leaving no choice to users about when to switch between modes, leading to a shorter coarse pointing phase than *Continuous* techniques.

(H2) *Head*-based techniques make mode switching cognitively less demanding, as they use different body parts to control the two modes. *Pad*-based techniques use the fingers in both modes *and* for mode switching. With *Head*-based techniques, a mode switch is triggered when the finger comes into contact with the pointing zone.

Related to (H1), we expect an effect of forcing the mode switch with *Discrete* techniques on the time spent in the precise phase, since this often entails engaging *Precise* pointing mode significantly farther away from the target than with *Continuous* techniques.

We also expect to observe relative differences between the two experiments due to the smaller pointing zone used in the second one: we expect *Head*-based techniques to be at an advantage in the second experiment, since the smaller size of the pointing zone will hinder performance of *Pad*-based techniques in *Coarse* mode.

General Design and Procedure

Apparatus. We used a wall-sized display consisting of 32 high-resolution 30" LCD panels (2560 × 1600 pixels each, 100 dpi, 1 mm ∼ 4 pixels) laid out in an 8 × 4 matrix, 5.5 meters wide and 1.8 meters high for a total resolution of 22080 × 7360 when including bezels (Figure 1) [5]. This wall display is driven by a cluster of 16 computers, each equipped with two high-end graphic cards (one per screen), and a master workstation. The experiments' software was implemented with jBricks [29]. We used a VICON motion-capture system to track passive IR retroreflective markers and provide the 3D

coordinates of the participants' head with sub-millimeter accuracy at 200Hz. Participants stood up, 2 meters away from the display. Given this position and the size of the wall, the operating range of the head was $\pi/2 \times \pi/5$ rad.

Trials. The task consisted in acquiring circular targets of width W. Participants first had to dwell for half a second in a dedicated zone in coarse mode, and could then acquire the target, positioned at a distance A from the cursor's position. Experimental conditions combined one of two target widths (W): 18 pixels (4.16 mm), 80 pixels (18.48 mm), and one of three movement amplitudes (A): 2760 pixels (width of one LCD panel, 0.637 m), 8280 pixels (width of 3 LCD panels, 1.912 m), 13 800 pixels (width of 5 LCD panels, 3.187 m).

Regarding the difficulty of these pointing tasks: according to Casiez *et al.* [10], the highest Fitts' Index of Difficulty (ID) tested prior to their article was 7.6 bits in a desktop context (A= 30 cm, W= 1.5 mm). In this same article they report a univariate (1D) Fitts' experiment with IDs as high as 9 (A= 4.5 m, W= 9 mm) on a 25ppi projected display. The highest ID tested in our studies is 9.54 using bivariate pointing tasks (2D) tasks. To our knowledge, such a level of difficulty was never evaluated before in this context.

For each movement amplitude A, targets were pseudo randomly positioned inside one of six LCD panels so that the average distance from the center of that panel to the target was about 600 pixels (150 mm). We chose this value so as to neither advantage nor disadvantage *Discrete* techniques (relative to the size of a screen), given that those techniques position the cursor at the center of the most recently visited panel.

Design and Procedure. Both experiments used a $4 \times 2 \times 3$ within-subject design with factors TECH (technique), W and A. We blocked by technique and used a Latin square to balance the presentation order of techniques among participants. At the beginning of each block of a *Head* technique, participants were asked to stand still and stare at the center of the display for 5 seconds to calibrate the center of their head's operating range, in order to balance the angular offsets described earlier. Each block started with a training session composed of two parts. In the first part, W and A were set to their largest values. The operator explained the technique and asked participants to practice until their performance stabilized, i.e. when they performed six consecutive trials with a movement time (*MT*) variation within 40% of their average. Participants were allowed to practice longer if they wanted to. The second training phase consisted of three blocks of six trials whose purpose was to put participants in conditions closer to the actual trials to be performed next: largest W and largest A, smallest W and medium A, smallest W and smallest A. The remaining TECH blocks were measured and decomposed into six sub-blocks composed of six replications of each of the W $\times$ A conditions described earlier. For both experiments, sessions lasted 40 minutes on average. At the end of a session, participants answered a questionnaire about their preferences and were encouraged to make comments.

Participants. The same 12 participants (10 men, 2 women; 24 to 38 year-old, avg. 29.6, med. 28.5) served in both experiments. All had normal or corrected-to-normal vision and were right-handed. All were daily users of personal computers and smartphones. Only two used tablets regularly.

Measures. We measure movement time *MT* and error rate. *MT* begins when participants start moving after having dwelled 0.5 s inside a circle centered on the start target, and stops at the first press event on target. We split *MT* into movement time of the coarse phase, *CMT*, and movement time of the precise phase, *PMT*, according to the time of the last switch to precise mode. We also wanted to measure the cost of mode switching. However finding a coherent measure of the duration of mode switching for all techniques is difficult: there is no explicit marker for the beginning of the switch with head-based techniques, since in coarse mode the finger is hovering over the device; switching to precise mode with Pad-based techniques can be performed in two ways, one of which is immediate (lifting one of the fingers). To evaluate the cost of mode switching, we measure *VPT*, the time between the last move event in coarse mode and the velocity peak in the subsequent precise phase. *VPT* is a "technique-agnostic" measure that provides a satisfying approximation of the time between coarse cursor stabilization and use of precise mode.

Tablet Experiment

In this first experiment, we used a tablet (Apple iPad I) with a resolution of 768×1024 pixels. Participants had to hold the tablet vertically. The pointing zone used the top 768×256 pixels (148×49 mm), resulting in an input resolution of 5.2 dot/mm. Its aspect ratio matched that of the wall display. Table 1 summarizes the CD gain parameters.

Quantitative-objective Results. We removed a few outliers: 0.69% of the trials had an unreasonable residual / predicted ratio. These outliers were mainly due to Wi-Fi transmission problems. As expected, *MT* distributions per condition are skewed. We thus perform our analyses using median values, per participant, on the model TECH $\times$ W $\times$ A $\times$ Rand(Participant). Figure 5 graphs *MT* for each technique[2].

An ANOVA reveals a significant effect of TECH on *MT* ($F_{3,33} = 11.4$, $p < 0.0001$). A post-hoc t-test with Bonferroni correction shows that $ContHead_L$ is significantly faster than all other techniques (all p's < 0.001): 9% faster than $ContPad_L$, 15% faster than $DiscPad_L$, and 14% faster than $DiscHead_L$. The only other significant difference is between $ContPad_L$ and $DiscPad_L$, the former being 6% faster.

We also observe, as expected, significant effects of target width W ($F_{1,11} = 459$, $p < 0.0001$) and movement amplitude A ($F_{2,22} = 43.8$, $p < 0.0001$) on *MT*, participants being faster with the larger target and the smallest distance. We do not observe any interaction effect TECH $\times$ W or TECH $\times$ A.

The overall error rate is 5.9%. An ANOVA reveals no effect of TECH on error rate ($F_{3,33} = 0.61$, $p = 0.6151$) and no significant interaction. Again, as expected, we find a significant effect of W on error rate ($F_{1,11} = 11.5$, $p = 0.0061$): 8.4% for small targets and 3.4% for larger ones. We also measure a significant effect of A ($F_{2,22} = 4.00$, $p = 0.0329$), although the

[2]In all our bar charts, the mean is taken over the medians of each experimental condition (including Participant). Error bars represent the corresponding 95% confidence limit.

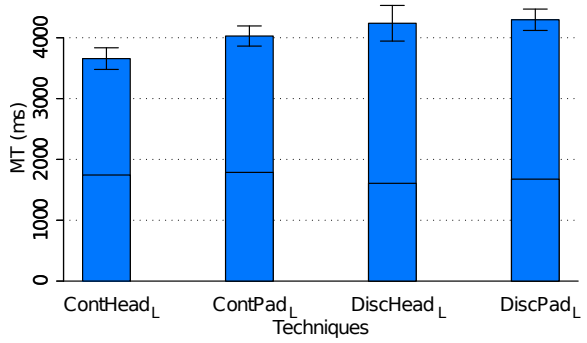


Figure 5. *MT* **for each** TECH **(tablet experiment). Black lines that split the four bars indicate the time of the last switch to precise mode.**

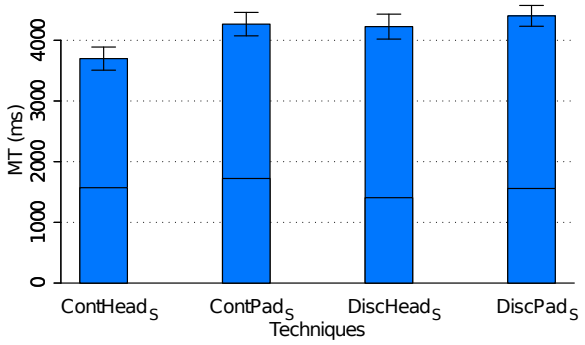


Figure 6. *MT* **for each** TECH **(smartphone experiment). Black lines that split the four bars indicate the time of the last switch to precise mode.**

magnitude of the difference is small: 4.9% for the largest amplitude, 6.9% and 5.9% for the medium and small ones.

As illustrated in Figure 5, the time spent in the coarse phase is slightly shorter with *Discrete* techniques than with *Continuous* techniques. However, this difference is not statistically significant – (H1) is not supported – and is not large enough to make *Discrete* techniques more efficient than *Continuous* ones. Indeed, the time spent in precise mode is far longer with the *Discrete* techniques. We tentatively attribute this to the fact that the last mode switch is performed 150 mm away from the target on average with *Discrete* techniques, compared to 67 mm with $ContHead_L$ and 71 mm with $ContPad_L$.

$ContHead_L$ and $ContPad_L$ feature very similar coarse pointing times (*CMT*) and distance-to-target at mode-switch time. $ContHead_L$'s shorter task time is mainly due to performance improvements during the precise pointing phase (Figure 5). We observed that velocity peaks occur earlier with $ContHead_L$ than with $ContPad_L$ (average *VPT* of 462 ms vs. 735 ms). Distances from the mode-switch location to the target are similar for *Head*- and *Pad*-based techniques, and their acceleration curves have the same input characteristics (V_{min}, V_{max} and $ratio_{inf}$). We thus expected the velocity peaks of $ContHead_L$ and $ContPad_L$ to occur at similar times. But participants actually required more time to reach full speed with $ContPad_L$. This suggests that the cost of mode switching is indeed lower for *Head*-based techniques than for *Pad*-based ones – supporting (H2) – possibly because of the cognitive cost entailed by switching between two very different CD gains whereas the input device / method remains the same.

Quantitative-subjective Results. Overall, participants preferred to use *Head*-based techniques (10 out of 12) and *Continuous* techniques (8 out of 12). 7 participants ranked $ContHead_L$ first, 4 ranked $DiscHead_L$ first and 2 ranked $ContPad_L$ first. However, there were no strong complaints about any particular technique, except for one participant who clearly stated that he disliked *Discrete* techniques.

Several participants complained about the lack of tactile feedback, which made it difficult to know when the fingers were leaving the pointing zone. They expressed the need for some sort of physical boundary to delimit the pointing zone, as featured by laptop touchpads. Only one participant answered that holding the tablet for 40 minutes was indeed a cause of fatigue when we inquired about this potential issue.

Smartphone Experiment

The second experiment used a smartphone (Apple iPod Touch) with a resolution of 480×320 pixels. Participants had to hold the device horizontally. The pointing zone used the top 480×166 pixels (75×26 mm, same aspect ratio as the display), resulting in an input resolution of 6.4 dot/mm. CD gain parameters are given in Table 1.

We took participants' comments about the lack of tactile feedback into account, and used easily removable tape to delimit the pointing zone (3 mm wide, 0.8 mm thick). Preliminary tests had shown that the tape did make it much easier to find the input zone without looking at the surface and to anticipate fingers crossing the pointing zone's boundaries.

Results are very similar to those of the tablet experiment reported above. We observe an effect of TECH on *MT* (no interaction with A and W), and an effect of A and W, as expected. (H2) is also supported by the data. We get similar error rates, and an overall subjective preference for $ContHead_S$. Movement time, split between the coarse and precise phases, is shown in Figure 6 using the same scale as in Figure 5.

Regarding the coarse pointing phase, this time we observe a significant effect of TECH ($F_{3,33} = 5.39$, $p = 0.0039$). $DiscHead_S$ is significantly faster than all other techniques, and $DiscPad_S$ and $ContHead_S$ are significantly faster than $ContPad_S$ (for this phase). With the smaller pointing zone used in this experiment, we do observe the hypothesized advantage for discrete techniques (H1). However, the time gained during the coarse phase is again not large enough to make *Discrete* techniques faster than *Continuous* ones.

Since we did not counterbalance the order of pointing zone sizes (all participants performed the tablet experiment first), and since we added tactile feedback using tape to delimit the zone in the second experiment, we cannot formally compare overall performance across both experiments. However, we can make three informal observations. First, the movement time difference between $ContHead_S$ and $ContPad_S$ is larger for the smartphone (13%) than for the tablet (9%); Second, $ContPad_S$ is not significantly faster than $DiscPad_S$, as opposed to $ContPad_L$ vs. $DiscPad_L$; Third, Figures 5 and 6 suggest that *DiscHead* performed better with the smartphone than with the tablet, relative to *ContPad* and *DiscPad*. These observations suggest that as we had anticipated, *Head*-based techniques are at an advantage with smaller pointing zones.

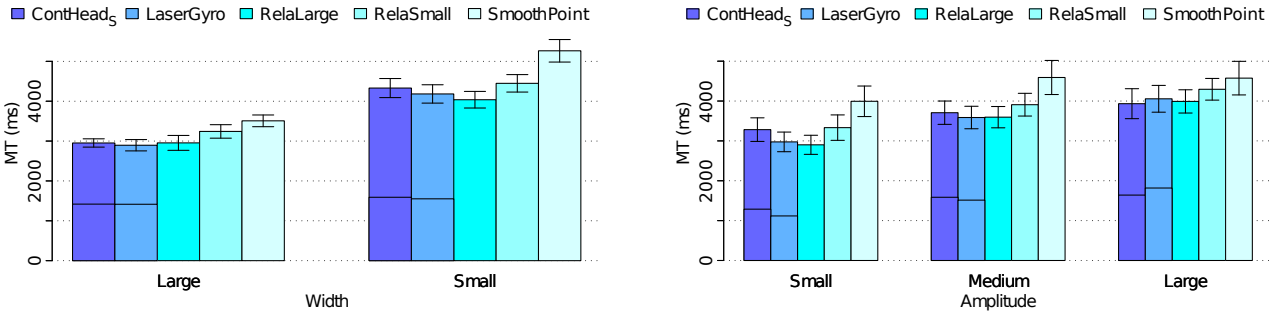


Figure 7. *MT* for each TECH by W and by A for the last experiment.

COMPARISON WITH STATE-OF-THE-ART TECHNIQUES

The above experiments identified *ContHead* as the fastest and preferred technique. As a follow-up, we wanted to evaluate the cost of using a handheld device for pointing. To this end, we ran a third experiment to compare *ContHead* to state-of-the-art remote pointing techniques. Such techniques do not meet requirement #2, i.e., accommodating widgets used for other purposes than pointing, but this comparison enables us to put our results in a more general context.

The comparison involves the version of *ContHead* that uses a small pointing zone, *ContHead*$_S$, on a tablet. *ContHead*$_S$ and *ContHead*$_L$ feature very similar performance (see Figures 5 and 6). This makes *ContHead*$_S$ a more interesting technique overall, since using a smaller pointing zone means that the device can accommodate more control widgets. We briefly describe the techniques selected for comparison below.

We noticed during early pilot studies that purely relative pointing techniques can actually be viable candidates for high-precision pointing, provided that their acceleration function is properly tuned. We thus decided to include purely relative, trackpad-like pointing in this experiment. We tested two different pointing zone sizes, equal to those of the pointing zones used in the previous experiments. For the purpose of experimental comparison, those were considered as distinct techniques, called *RelaSmall* and *RelaLarge*, each tuned with its own optimized transfer function. As discussed before, the literature on calibrating transfer functions for relative pointing techniques is scarce. The functions used by major operating systems do not meet the requirements of ultra-high-resolution wall-sized displays in terms of speed and precision. We therefore used the transfer function calibration method initially developed for our dual-precision techniques. Parameter values are given in Table 1. Compared to the functions used in the first two experiments, these two functions feature a much higher CD_{max} that allows traveling much larger distances without too much clutching.

To avoid introducing a confound due to the device itself, we decided to use a tablet for all techniques that required a handheld device: *ContHead*$_S$, *RelaSmall* and *RelaLarge*. We compared those three techniques with two techniques from the literature: *LaserGyro* and *SmoothPoint*.

LaserGyro is a mid-air dual-precision technique that provides coarse control of the cursor's position with ray-casting and enables precise adjustments with relative angular motion. Mode switches are triggered with a button. This technique, inspired by Vogel *et al.*'s work on distant freehand pointing [34], has been shown to perform very well [26].

SmoothPoint [15] also combines ray-casting with relative pointing. However, the transition between the two modes is progressive, based on a transfer function that depends on input motion velocity. The authors of [15] propose a method to tune this function, but pilot tests in our environment revealed that this method does not scale to large surfaces with high pixel densities such as those typically encountered on ultra-high-resolution wall displays: the difference between the minimum and maximum CD gain values is too high, causing the precise mode to be very jerky when selecting small targets such as those considered here. We transposed the calibration method described in [15] to our context[3] to the best of our abilities, iterating until the technique eventually enabled us to achieve pointing tasks with the target sizes and movement amplitudes considered in our experiments.

In this third experiment, we tested a total of five techniques (TECH): *ContHead*$_S$, *RelaSmall*, *RelaLarge*, *SmoothPoint* and *LaserGyro*. The apparatus, design, and procedure were exactly the same as in the previous experiments. We added a physical border around the large and small pointing zones on the tablet to limit the need to look at the input device by providing tactile feedback when the fingers were about to leave the zone. We used a 5×5 latin square to balance technique ordering. 15 participants served in the experiment (10 of them had participated in the previous experiments 1.5 months before, 5 were new and assigned to the same Latin square).

Quantitative-objective Results. As in the previous experiment we removed a few outliers (1.28%). Figure 7 shows movement time *MT* by W and A. For *ContHead*$_S$ the results are very close to those of the previous experiments. We observe that *SmoothPoint* performs poorly compared to all other techniques (significantly so, for each W and A condition). For the sake of conciseness, we do not report figures for *SmoothPoint* in post-hoc tests, even though it was, of course, included in the corresponding statistical computations.

An ANOVA reveals a significant effect of TECH on *MT* ($F_{4,56} = 20.5, p < 0.0001$). A post-hoc t-test with Bonferonni correction shows that *ContHead*$_S$, *LaserGyro* and *RelaLarge* are all significantly faster than *RelaSmall* (with speed-ups of 5.4%, 8% and 9%, respectively).

[3]We did not use our logistic function because the transfer function associated with SmoothPoint is part of the technique [15].

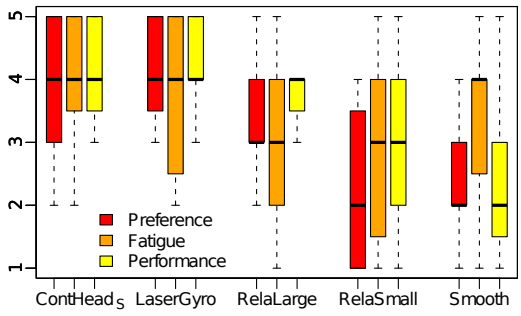


Figure 8. Preference, fatigue and (self-reported) perceived performance for each TECH on a five-point Likert item (5 is best, 1 is worst).

This time, the ANOVA reveals significant interactions TECH × W ($F_{4,56} = 5.90$, $p = 0.0005$) and TECH × A ($F_{8,112} = 2.60$, $p = 0.0122$) on *MT*. One cause for these interactions is *SmoothPoint*, which is slower for small targets than for larger ones, and faster for the largest amplitude. We can also observe (Figure 7) that the two relative techniques, *RelaSmall* and *RelaLarge*, are faster for the small width and the small amplitude. Indeed, post-hoc tests show that (i) for small targets, the only significant result is that *RelaLarge* is faster than *RelaSmall*, and for large targets, *ContHead*$_S$ and *LaserGyro* are also significantly faster than *RelaSmall*; (ii) for the small amplitude the only significant results are that *RelaLarge* is faster than *RelaSmall* and *ContHead*$_S$, while for the large amplitude the only significant results are that both *RelaLarge* and *ContHead*$_S$ are significantly faster than *RelaSmall*.

The overall error rate is 9.6%. *ContHead*$_S$, *RelaSmall* and *RelaLarge* feature low error rates (3.90%, 3.52% and 4.04%) with only marginal differences between large and small targets (2.5% vs 5.17%). *LaserGyro* and *SmoothPoint* also feature low error rates for large targets (0.70% and 5.60%), but the error rate rises dramatically for small targets: 25.6% and 43.0%. We had not anticipated such an increase in error rate from our pilot studies. The problem turned out to come from the fact that clicking with the handheld wireless mouse caused small hand movements which, in turn, caused small cursor displacements that were sometimes large enough to make the cursor leave the target. We furthered our analysis by measuring the time to first click (instead of first click on the target). This did not change the results for *LaserGyro*. However, results were slightly different for *SmoothPoint*, which remained slower than *RelaSmall* in all conditions but not significantly so for the small target + large amplitude condition.

Quantitative-subjective Results. At the end of the experiment, we asked participants to rank the techniques on a five-point Likert scale in terms of preference, fatigue and perceived performance. Figure 8 summarizes the results in a boxplot. Kruskal-Wallis tests revealed a significant effect of TECH on Preference ($\chi^2_4 = 28.5$, $p < 0.0001$) and Performance ($\chi^2_4 = 23.4$, $p = 0.0001$), but not on Fatigue. Post-hoc tests using Mann-Whitney tests with Bonferroni correction show that (i) *ContHead*$_S$ and *LaserGyro* were preferred to both *RelaSmall* and *SmoothPoint*; (ii) *RelaLarge* was preferred to *SmoothPoint*; (iii) *ContHead*$_S$, *LaserGyro* and *RelaLarge* were perceived as faster than *SmoothPoint*; and (iv) *LaserGyro* and *ContHead*$_S$ were perceived as (marginally) faster than *RelaLarge*.

Discussion and Design Guidelines

Results show that using a handheld device for high precision pointing enables users to point at least as efficiently as state-of-the-art mid-air pointing techniques in terms of both speed and accuracy, while leaving a large portion of the handheld's surface available for additional, application-specific widgets.

The three techniques that perform best in terms of movement time and preference are *RelaLarge*, *ContHead* and *LaserGyro*. While there is no significant speed difference between them, each technique has its own strengths and limitations, making it suitable to specific contexts of use.

While relative pointing is not novel, making it work efficiently in such challenging contexts is an interesting result. Indeed, existing functions, even elaborate ones such as that of *SmoothPoint*, were designed for lower-resolution environments, and fare poorly with the high Fitts' IDs considered here. *RelaLarge*, based on the CD transfer function tuning method introduced in this paper, provides pointing performance that matches that of more elaborate techniques. It is straightforward to implement and does not require specific equipment to track spatial position and orientation.

RelaSmall is also quite an achievement, considering that it provides enough precision to perform bivariate pointing tasks of high difficulty (up to 9.5) with a pointing zone of 20 cm^2 only (approximately $1/4^{th}$ of *RelaLarge*'s surface area). Given that the pointing zone dimensions of *RelaLarge* preclude its use on smartphones, *RelaSmall* can be seen as an interesting option. Indeed, the technique will only incur a 5-to-10-percent performance decrease compared to more efficient techniques, which can be considered an acceptable tradeoff when only small handheld devices are available, or when a large portion of the handheld's screen real-estate should be allocated to additional interface widgets.

This performance decrease can be avoided by using *ContHead*, which achieves the same level of performance as *RelaLarge* but on a much smaller input area, equivalent to that of *RelaSmall*. *ContHead* should be considered when the tasks and context of use require many additional interface widgets or when only smartphones are available, provided that tracking the location and orientation of the head is possible.

Finally, *LaserGyro* causes many more errors than other techniques for small values of W, as the tremor caused by pressing a physical button, even if comparatively small, is large enough to severely hinder acquiring very small targets. This problem does not happen with tablet-based techniques since their tapping mechanism is algorithmically decoupled from their pointing mechanism. Another drawback of this technique is that it cannot accommodate additional interface widgets on the input device. However, *LaserGyro* leaves the non-dominant hand free to perform other interactions, making it a relevant option when the task requires operating additional input devices, provided that pointing task IDs are lower than 9.

CONCLUSION AND FUTURE WORK

This paper investigated the use of handheld devices for very difficult remote pointing tasks on wall displays, where only a portion of the handheld's surface is dedicated to pointing. The goal is to leave the larger part of this surface to task-specific

widgets, since pointing is rarely sufficient on its own to interact with wall displays. We designed and evaluated techniques that use various input channels to improve pointing at very small targets across large amplitudes.

Our most successful design, *ContHead*, lets users perform pointing actions at two levels of granularity: coarse pointing uses the natural movements of the head when moving the cursor across large distances; precise pointing uses a small pointing zone on the handheld device to perform relative pointing movements via finger gestures. When compared with state-of-the-art techniques that are not constrained by our requirements, the technique performs well even with high indices of difficulty, beyond those tested in previous work. *ContHead* was also rated as one of the best techniques overall in terms of subjective preference and perceived performance.

Purely relative techniques using only the touch surface of the handheld device performed better than we had originally anticipated. This led us to carefully investigate Control-Display transfer functions that enable both very fast pointing across large amplitudes while minimizing clutching, and precise cursor adjustments to acquire targets only a few millimeters in diameter. We showed that with a large-enough pointing zone and proper tuning, such a relative technique competes with the most efficient dual-mode techniques.

In future work we plan to investigate how to automatically couple and decouple cursor position from head orientation since it is not always desirable, in real contexts of use, to have them tied together. One possible solution would be to detect situations where the user is likely to initiate pointing actions, e.g., using information about the handheld device and the position of the hand [31, 35].

REFERENCES

1. Ashdown, M., Oka, K., and Sato, Y. Combining head tracking and mouse input for a GUI on multiple monitors. In *Proc. CHI EA '05*, ACM (2005), 1188–1191.
2. Ball, R., North, C., and Bowman, D. A. Move to improve: promoting physical navigation to increase user performance with large displays. In *Proc. CHI '07*, ACM (2007), 191–200.
3. Baudisch, P., Cutrell, E., Robbins, D., Czerwinski, M., Tandler, P., Bederson, B., and Zierlinger, A. Drag-and-Pop and Drag-and-Pick: Techniques for Accessing Remote Screen Content on Touch and Pen-operated Systems. In *Proc. INTERACT '03* (2003), 57–64.
4. Baudisch, P., Sinclair, M., and Wilson, A. Soap: a pointing device that works in mid-air. In *Proc. UIST '06*, ACM (2006), 43–46.
5. Beaudouin-Lafon, M., Chapuis, O., Eagan, J., Gjerlufsen, T., Huot, S., Klokmose, C., Mackay, W., Nancel, M., Pietriga, E., Pillias, C., Primet, R., and Wagner, J. Multi-surface interaction in the WILD room. *IEEE Computer 45*, 4 (2012), 48–56.
6. Bezerianos, A., and Balakrishnan, R. The Vacuum: facilitating the manipulation of distant objects. In *Proc. CHI '05*, ACM (2005), 361–370.
7. Blanch, R., and Ortega, M. Rake cursor: improving pointing performance with concurrent input channels. In *Proc. CHI '09*, ACM (2009), 1415–1418.
8. Boring, S., Baur, D., Butz, A., Gustafson, S., and Baudisch, P. Touch projector: mobile interaction through video. In *Proc. CHI '10*, ACM (2010), 2287–2296.
9. Casiez, G., and Roussel, N. No more bricolage!: methods and tools to characterize, replicate and compare pointing transfer functions. In *Proc. UIST '11*, ACM (2011), 603–614.
10. Casiez, G., Vogel, D., Balakrishnan, R., and Cockburn, A. The impact of control-display gain on user performance in pointing tasks. *HCI 23*, 3 (2008), 215–250.
11. Forlines, C., Vogel, D., and Balakrishnan, R. Hybridpointing: fluid switching between absolute and relative pointing with a direct input device. In *Proc. UIST '06*, ACM (2006), 211–220.
12. Freedman, E. G., and Sparks, D. L. Coordination of the eyes and head: movement kinematics. *Exp. Brain Res. 131* (2000), 22–32.
13. Frees, S., Kessler, G. D., and Kay, E. PRISM interaction for enhancing control in immersive virtual environments. *ACM ToCHI 14*, 1 (2007).
14. Gallo, L., Ciampi, M., and Minutolo, A. Smoothed pointing: A user-friendly technique for precision enhanced remote pointing. In *Proc. CISIS '10*, IEEE (2010), 712–717.
15. Gallo, L., and Minutolo, A. Design and comparative evaluation of smoothed pointing: A velocity-oriented remote pointing enhancement technique. *IJHCS 70*, 4 (2012), 287–300.
16. Jacob, R. The use of eye movements in human-computer interaction techniques: what you look at is what you get. *ACM TOIS 9* (1991), 152–169.
17. Jota, R., Nacenta, M. A., Jorge, J. A., Carpendale, S., and Greenberg, S. A comparison of ray pointing techniques for very large displays. In *Proc. GI '10*, CIPS (2010), 269–276.
18. Kirstein, C., and Muller, H. Interaction with a projection screen using a camera-tracked laser pointer. In *Proc. MMM '98*, IEEE (1998), 191–192.
19. Kitajima, K., Sato, Y., and Koike, H. Vision-based face tracking system for window interface: prototype application and empirical studies. In *Proc. CHI EA '01*, ACM (2001), 359–360.
20. Kollerl, D. R., Mine, M. R., and Hudson, S. E. Head-tracked orbital viewing: an interaction technique for immersive virtual environments. In *Proc. UIST '96*, ACM (1996), 81–82.
21. Konig, W. A., Gerken, J., Dierdorf, S., and Reiterer, H. Adaptive pointing - design and evaluation of a precision enhancing technique for absolute pointing devices. In *Proc. INTERACT '09*, Springer (2009), 658–671.
22. Kopper, R., Bowman, D. A., Silva, M. G., and McMahan, R. P. A human motor behavior model for distal pointing tasks. *IJHCS 68*, 10 (2010), 603–615.
23. Mccallum, D. C., and Irani, P. Arc-pad: absolute+relative cursor positioning for large displays with a mobile touchscreen. In *Proc. UIST '09*, ACM (2009), 153–156.
24. Morency, L.-P., and Darrell, T. Head gesture recognition in intelligent interfaces: the role of context in improving recognition. In *Proc. IUI '06*, ACM (2006), 32–38.
25. Myers, B. A., Bhatnagar, R., Nichols, J., Peck, C. H., Kong, D., Miller, R. C., and Long, A. C. Interacting at a distance: measuring the performance of laser pointers and other devices. In *Proc. CHI '02*, ACM (2002), 33–40.
26. Nancel, M., Pietriga, E., and Beaudouin-Lafon, M. Precision pointing for ultra-high-resolution wall displays. Technical Report RR-7624, INRIA, 2011.
27. Nancel, M., Wagner, J., Pietriga, E., Chapuis, O., and Mackay, W. Mid-air pan-and-zoom on wall-sized displays. In *Proc. CHI '11*, ACM (2011), 177–186.
28. Nickel, K., and Stiefelhagen, R. Pointing gesture recognition based on 3d-tracking of face, hands and head orientation. In *Proc. ICMI '03*, ACM (2003), 140–146.
29. Pietriga, E., Huot, S., Nancel, M., and Primet, R. Rapid development of user interfaces on cluster-driven wall displays with jBricks. In *Proc. EICS '11*, ACM (2011), 185–190.
30. Poupyrev, I., Tomokazu, N., and Weghorst, S. Virtual Notepad: Handwriting in Immersive VR. In *Proc. VRAIS '98*, IEEE (1998), 126–132.
31. Rahman, M., Gustafson, S., Irani, P., and Subramanian, S. Tilt techniques: Investigating the dexterity of wrist-based input. In *Proc. CHI '09*, ACM (2009), 1943–1952.
32. Räihä, K.-J., and Špakov, O. Disambiguating ninja cursors with eye gaze. In *Proc. CHI '09*, ACM (2009), 1411–1414.
33. Stahl, J. S. Amplitude of human head movements associated with horizontal saccades. *Exp. Brain Res. 126* (1999), 41–54.
34. Vogel, D., and Balakrishnan, R. Distant freehand pointing and clicking on very large, high resolution displays. In *Proc. UIST '05*, ACM (2005), 33–42.
35. Wagner, J., Huot, S., and Mackay, W. Bitouch and bipad: Designing bimanual interaction for hand-held tablets. In *Proc. CHI '12*, ACM (2012).
36. Zhai, S., Morimoto, C., and Ihde, S. Manual and gaze input cascaded (MAGIC) pointing. In *Proc. CHI '00*, ACM (1999), 246–253.